The MIT Press

SPASM, a Real-Time Vocal Tract Physical Model Controller; and Singer, the Companion Software Synthesis System
Author(s): Perry R. Cook
Source: *Computer Music Journal*, Vol. 17, No. 1 (Spring, 1993), pp. 30-44
Published by: The MIT Press
Stable URL: http://www.jstor.org/stable/3680568
Accessed: 09/09/2013 15:36

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at
http://www.jstor.org/page/info/about/policies/terms.jsp

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

The MIT Press is collaborating with JSTOR to digitize, preserve and extend access to *Computer Music Journal*.

http://www.jstor.org

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Perry R. Cook
Stanford Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305 USA
prc@ccrma.stanford.edu

SPASM, a Real-Time Vocal Tract Physical Model Controller; and Singer, the Companion Software Synthesis System

The synthesis of singing has been investigated from within the frameworks of most music synthesis methods. Primitive speech systems use time-domain methods such as playback of digitally recorded phonemes or words, but such systems are generally inappropriate for the creative process of music composition and synthesis. Many systems are based on the final spectrum (McAulay and Quatieri 1986; Chowning 1989; Serra and Smith 1990). Some systems include parameters for controlling formants or other spectral features directly (Rodet 1984). Others—specifically source/filter systems such as linear predictive coding (LPC) (Atal 1970)—are more closely related to the physics of the vocal tract. Such systems often allow analysis/resynthesis, but the mapping of such nonphysical or pseudo-physical systems onto the actual physical components of the vocal mechanism is often not sufficient to allow direct synthesis based on notions from vocal pedagogy and speech physiology such as tongue position and glottal effort.

Toward the goal of creating a system that can be controlled from such physical descriptions, a model of the vocal tract filter was developed using the waveguide formulation (Smith 1987). Two voice synthesis systems have been constructed using this physically parameterized model of the human vocal tract. One system is a real-time digital signal processor (DSP) interface program, which allows graphical interactive experimentation with the various control parameters. The other is a text-driven software synthesis program. The vocal tract is modeled in both systems by a digital waveguide filter (WGF), which is controlled directly by shape parameters. A nasal tract WGF is coupled to the vocal tract at the point where the velum controls the flow of air and sound into and from the nasal passages. Glottal source pulses are stored and retrieved from multiple wavetables. A filtered pulsed noise component is added to the periodic glottal source, simulating the turbulence that is generated as air flows through the oscillating vocal folds. To simulate the turbulences of fricatives and other consonants, a filtered noise source with up to two resonant frequencies can be placed at any point within the vocal tract.

The real-time DSP program is called SPASM (Singing Physical Articulatory Synthesis Model). The vocal tract shape is graphically displayed by a cross section of a human head. Sliders on an editor window control the radius of each vocal tract segment, the size of the velum opening into the nasal tract, and the radius of each nasal tract segment. A formant editor window displays the log-magnitude frequency response of the vocal tract. The glottal pulse shape can be edited in the time domain, and the spectrum can be edited in the frequency domain. Other real-time controls allow experimentation with pitch and vibrato. The system can interactively record a consonant and design a matching filter for use in resynthesis. Similarly, a vowel can be recorded, and the periodic glottal waveform and noise can be identified and stored. All settings can be saved as files.

The software synthesis system called "singer," takes as input a file of C-language function calls that specify the events to be synthesized. These function calls can be viewed as a time-ordered event list for controlling the vocal model. An event specification

Computer Music Journal, Vol. 17, No. 1, Spring 1993,
© 1993 Massachusetts Institute of Technology.

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

includes a transition time, shape and glottal files as written out by the SPASM system, noise and glottal volumes, glottal frequency (in hertz, as a musical note name, or as C expressions), and vibrato amount. The system synthesizes a sound file, smoothly interpolating from each set of parameters to the next over the times specified. Interpolations between all parameters of shape, glottal input, and noise filter control are on the single-sample level. In this way smoothly varying connected singing performances can be generated. All other parameters, such as random vibrato amount and periodic vibrato rate can be changed at any time but, for computational speed, are not recalculated for each sample.

A Physical Model of the Vocal Tract

Modeling the Acoustic Tube as a Waveguide Network

The waveguide filter (WGF) description of one-dimensional wave propagation in a tube provides the framework for controlling the vocal tract filter directly from physical measurements (Smith 1987). Smith (1992) has already outlined the mathematics of the waveguide formulation for strings, and the parallel derivation for the acoustic tube will be presented here. The tract is divided into a number of sections, each of the same length determined by the sampling rate and the speed of sound. The equations expressing conservation of momentum and mass are,

$$a(x)\frac{\partial P(x,t)}{\partial x} = -\rho\frac{\partial U(x,t)}{\partial t} \text{ and} \tag{1}$$

$$\frac{\partial U(x,t)}{\partial x} = -\frac{a(x)}{c^2}\frac{\partial P(x,t)}{\partial t}, \tag{2}$$

where $a(x)$ is the tube cross-sectional area at position x, ρ is the density of air, c is the velocity of sound, and $P_m(x,t)$ and $U_m(x,t)$ are the pressure and volume velocity at position x at time t. Webster's horn equation can then be derived to describe wave behavior in the tube,

$$\frac{\partial}{\partial x}\left[\frac{1}{a(x)}\frac{\partial U(x,t)}{\partial x}\right] = \frac{1}{c^2 a(x)}\frac{\partial^2 U(x,t)}{\partial t^2}. \tag{3}$$

In the sampled acoustic tube case, $a_m(x)$ is constant within each section m, that is, $a_m(x) = a_m$, then Webster's horn equation reduces to the wave equation within each section,

$$\frac{\partial^2 U_m(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 U_m(x,t)}{\partial t^2} \tag{4}$$

$$\frac{\partial^2 P_m(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 P_m(x,t)}{\partial t^2} \tag{5}$$

The solution of Eq. (5) can be expressed as a decomposition of leftgoing and rightgoing traveling pressure (or, optionally, velocity) waves,

$$P_m(x,t) = P_m^+\left(t - \frac{x}{c}\right) + P_m^-\left(t + \frac{x}{c}\right), \tag{6}$$

where P_m^+ and P_m^- are the rightgoing and leftgoing pressure wave components, respectively. This yields an expression of the wave equation in leftgoing and rightgoing wave variables,

$$\frac{\partial^2 P_m(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\left[P^+\left(t - \frac{x}{c}\right) + P^-\left(t + \frac{x}{c}\right)\right]. \tag{7}$$

To relate pressure to velocity, the expression

$$\left[\frac{\partial P_m^+}{\partial x} + \frac{\partial P_m^-}{\partial x}\right] = \frac{\rho c}{a_m}\left[\frac{\partial U_m^+}{\partial x} + \frac{\partial U_m^-}{\partial x}\right] \tag{8}$$

can be derived. Define the characteristic impedance R_m of tube section m as

$$R_m \equiv \frac{\rho c}{a_m}. \tag{9}$$

By integrating both sides and ignoring any constant terms as acoustically insignificant, the expressions

$$\left[P_m^+ + P_m^-\right] = R_m\left[U_m^+ - U_m^-\right], \tag{10}$$

$$P_m^+ = R_m U_m^+, \text{ and} \tag{11}$$

$$P_m^- = -R_m U_m^- \tag{12}$$

can be derived, which relate pressure and velocity

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 1. A smooth tube, the sampled equivalent, and the digital waveguide filter realization.

within the tube. Whenever two sections of tube meet, the boundary conditions to be satisfied are conservation of mass (volumetric flow is conserved) and conservation of momentum (pressure is continuous at the junction). These two conditions yield the junction scattering relations:

$$P_1^- = \frac{R_2 - R_1}{R_2 + R_1} P_1^+ + \left(1 - \frac{R_2 - R_1}{R_2 + R_1}\right) P_2^- \quad (13)$$

$$P_1^- = \left(1 + \frac{R_2 - R_1}{R_2 + R_1}\right) P_1^+ - \frac{R_2 - R_1}{R_2 + R_1} P_2^- \quad (14)$$

By defining the junction scattering coefficient k as

$$k = \frac{R_2 - R_1}{R_2 + R_1}, \quad (15)$$

the scattering relations for pressure and velocity at a junction can be expressed as

$$P_1^- = kP_1^+ + (1 - k)P_2^- \quad (16)$$

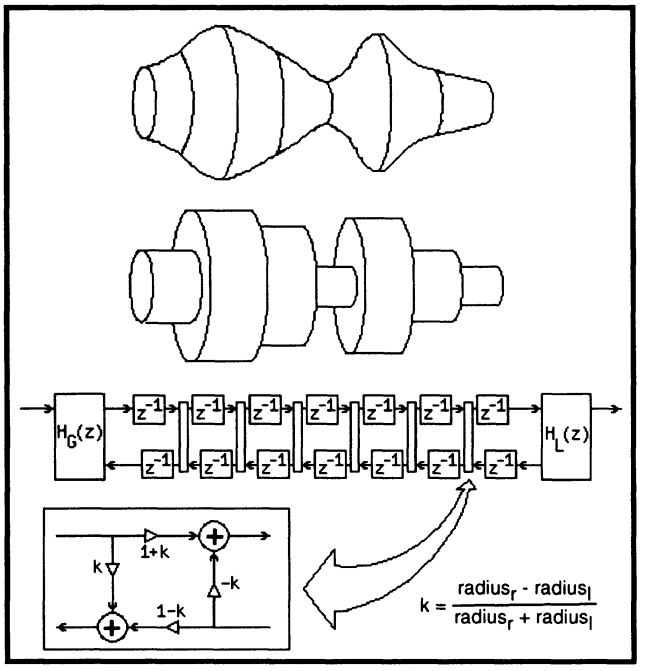
$$P_2^+ = (1 + k)P_1^+ - P_2^- \quad (17)$$

$$U_1^- = kU_1^+ + (1 + k)U_2^- \quad (18)$$

$$U_2^+ = (1 - k)U_1^+ - U_2^- \quad (19)$$

Figure 1 shows a smooth acoustic tube, the sampled space equivalent, and the digital WGF realization.

Since the characteristic impedance of each tube section is a function of the cross-sectional area of a section, and thus the radius, the junction scattering coefficients can be computed entirely from the physical tract section measurements. For the human voice, block $H_G(z)$ in Fig.1 represents the transmission and reflection characteristics of the glottis. The reflection characteristic of the glottis can be modeled most simply as a constant positive (for pressure) reflection coefficient ($\leq$=1), or more elaborately as a time-varying filter. $H_L(z)$ represents the reflection and transmission characteristics of the lip, which vary with the configuration of the vocal tract. The transmission and reflection functions should be complementary; that is, in a lossless system, any energy not reflected at the lips should be transmitted. A simple model of the lip reflection filter is a low-order



low-pass filter. Figure 2 shows three configurations of the single acoustic tube vocal tract model, with the log-magnitude frequency response of the WGF. The frequencies of the first three formant resonances are noted. The vocal tract was constructed using nine segments, modeled after the dimensions of a small female vocal tract (14 cm length) sampled at 22,050 Hz sampling rate.

Using only a single acoustic tube, this filter can be reduced to various lattice filter forms (Kelly and Lochbaum 1962; Markel and Gray 1976), or, given specific assumptions, to a canonical all-pole filter as is often used in LPC (Atal 1970). The WGF formulation, however, provides a clear picture of wave motion within the tube, simultaneous and instantaneous access to energy across both space and time, the ability to insert distributed losses, and, most importantly, a structure for connecting multiple waveguide paths. This allows the nasal tract to be simulated with another lattice filter, with the two tracts coupled at a bifurcation point located at the velum opening with a three-way scattering junction.

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 2. Three vocal tract shapes, the frequency response of the corresponding digital filter realization, and the frequencies of the first three formants.

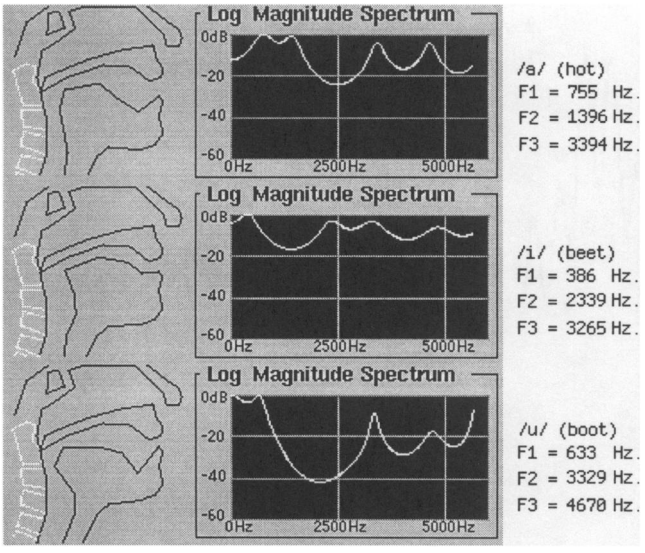
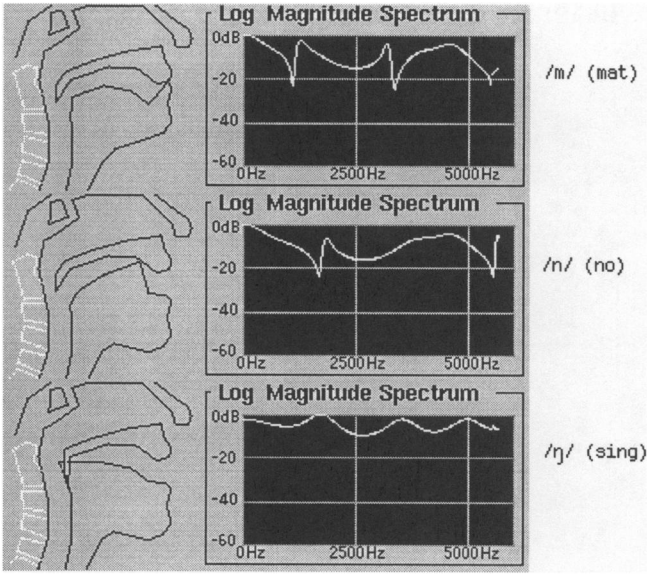


Fig. 3. Vocal tract shapes and the corresponding log-magnitude transfer functions for three nasals.



N-Way Scattering: Coupling the Nasal Tract to the Vocal Tract

The boundary conditions of pressure continuity and flow conservation determine the relationship between pressure and volume velocity at the junction of any number of waveguide tubes. Given a junction where n tubes meet, there are n incoming waves and tube impedances that are known, as well as n outgoing waves to be calculated. Denote the incoming pressure and velocity waves in tube i as P_i^+ and U_i^+, and the outgoing waves in tube i as P_i^- and U_i^-. Pressure and velocity are related according to Eqs. (10–12). The boundary conditions are

$$P_1 = P_2 = P_3 = \cdots = P_n = P_J \text{ and} \tag{20}$$

$$U_1 + U_2 + U_3 + \cdots + U_n = 0, \tag{21}$$

where P_J is the junction pressure. The characteristic admittance of the ith tube is the inverse of its characteristic impedance,

$$\Gamma_i = \frac{1}{R_i}. \tag{22}$$

It can be shown that

$$P_J = \frac{2\sum_{i=1}^{n} \Gamma_i P_i^+}{\sum_{i=1}^{n} \Gamma_i} \tag{23}$$

$$= \sum_{i=1}^{n} \alpha_i P_i^+, \tag{24}$$

where

$$\alpha_i = \frac{2\Gamma_i}{\sum_{i=1}^{n} \Gamma_i} \in [0,2] \tag{25}$$

Since $P_i = P_J$ and $P_i = P_i^+ + P_i^-$ for all i, the reflected pressure in any tube is simply the difference between the incoming pressure from that tube and the junction pressure.

$$P_i^- = P_J - P_i^+. \tag{26}$$

The reflected volume velocity is given by the product of the characteristic impedance of the tube and

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

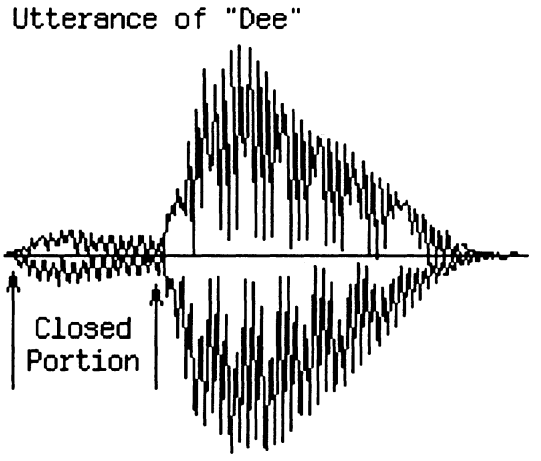
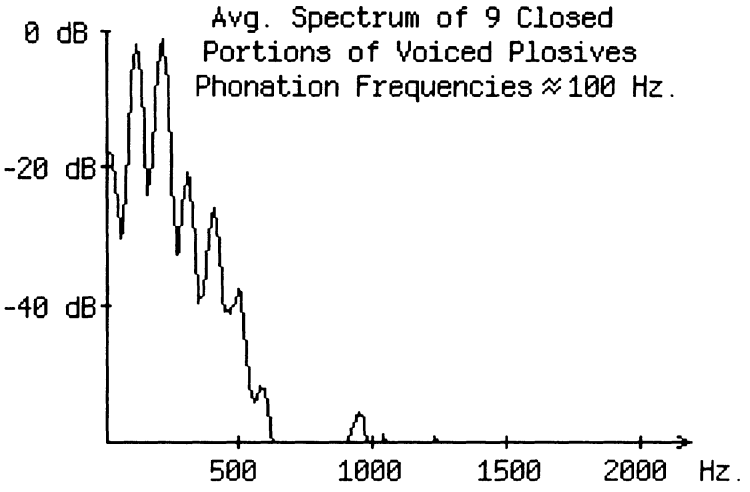


Fig. 4. One utterance containing a voiced plosive, and the average spectrum of nine closed voiced portions.

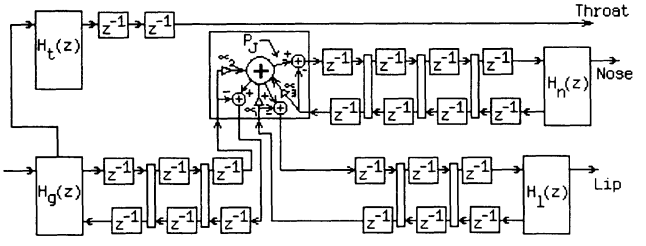


the reflected pressure. Figure 3 shows the vocal tract configurations and transfer functions of three nasals.

One More Path Out: Transcutaneous Radiation

A small but significant amount of acoustic energy is radiated out through the throat wall, especially in cases of voiced plosives and at times when all other paths out of the vocal tract are closed. Figure 4 shows the envelope of the utterance of a voiced plosive, and the spectrum of the radiation from the throat of a male speaker during the closed portion of a number of voiced plosives. The microphone was placed 6 inches from the lips, and the speaker uttered the sounds "Bee, Dee, Gee, Boo, Doo, Goo, Bah, Dah, Gah." Spectra were calculated for the closed portions of all nine utterances, then averaged. The average power of the closed portions of the utterances was –13.9 dB referenced to the average power of the open vowel portion of the utterances. The spectrum shows a low passed response and suggests that the throat radiation can be modeled to a good approximation by a low-order recursive low-pass filter. To account for group delay effects, a delay line can be added to simulate the length of the sound path from the throat to the point in space where the mixed vocal sound is desired. Figure 5 shows the digital filter real-

Fig. 5. A digital realization of the vocal tract, showing waveguides for the oral and nasal passages, the three-way scattering junction at the velum, and the low-pass filter and delay line that model radiation from the throat.



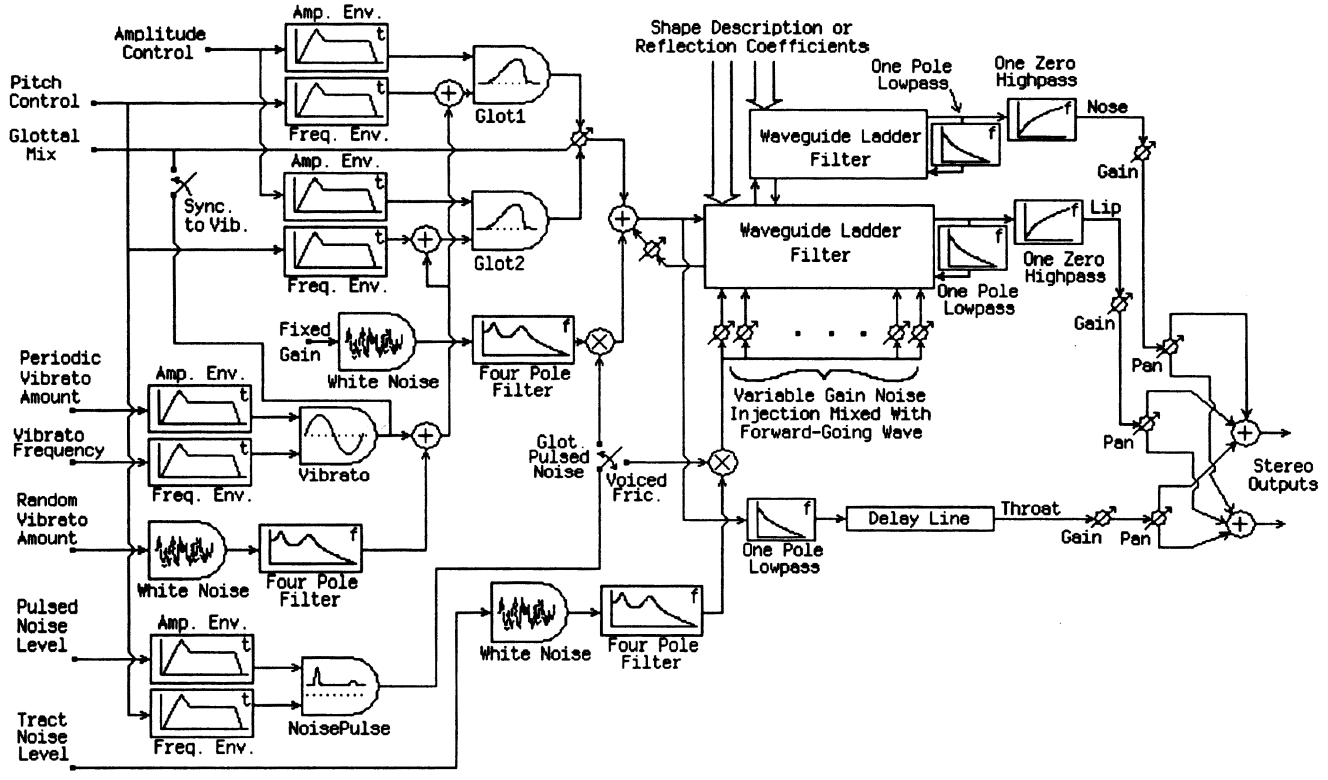
ization of the vocal tract, including the three-way scattering junction at the velum. $H_n(z)$ is the reflection/transmission filter for the nose, which is fixed under normal speaking and singing conditions.

The Full Model

Figure 6 shows the full vocal synthesis model. Two glottal source wavetable oscillators are provided for modulation effects and smooth interpolations from one vocal quality to another. Random and periodic vibrato oscillators and a pitch envelope generator provide frequency control to the glottal source oscillators. The glottal folds can produce an oscillating source of turbulent noise, especially at low frequencies (Cook 1991b). This effect is modeled by a pulsed filtered noise source mixed with the glottal oscillators.

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 6. Schematic of the complete vocal synthesis model.



The SPASM System

Design Goals

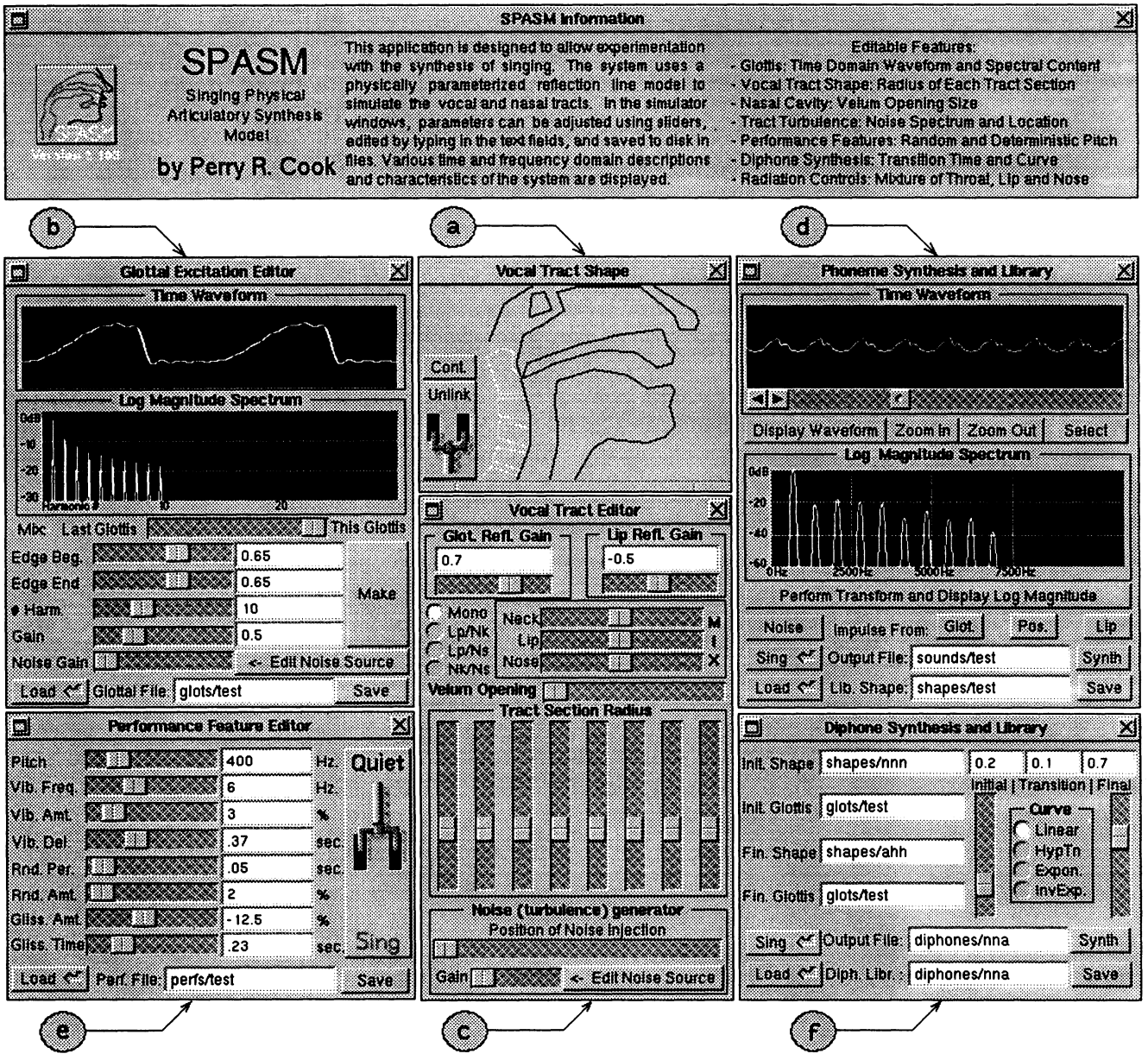
The SPASM (Singing Physical Articulatory Synthesis Model) system was developed in order to place the waveguide vocal tract model into a graphical interactive environment for experimentation and synthesis. The following three design goals were adhered to. (1) The main goal of the system is to produce high-quality musical vocal synthesis. (2) Some knowledge of music and musical acoustics on the part of the target user is assumed, but no knowledge of digital filter design or other engineering topics should be necessary to use the system. Technically experienced users should be able to access advanced features. (3) Model characteristics should be selected such that, wherever possible, control parameters are physically meaningful. Goals (1) and (2) take precedence over this requirement.

The Main System Screens

Figure 7 shows the main system screen. The various windows allow the user to modify the parameters controlling the model. Displays show shape, time, and spectral descriptions of the model and its signals. The windows that are visible when the program is first run are those that are required to do initial synthesis experiments. Figure 7a is the *Vocal Tract Shape Editor* window, which controls and displays the shape of the vocal tract. Figure 7b is the *Glottal Excitation Editor* window, which allows time and frequency-domain control of the glottal source, and saving of glottal parameter files to disk. Figure 7c is the *Noise Generator* controller, which controls placement of the noise source within the tract and the gain of the injected noise and provides access to a noise source filter editor. Figure 7d is the *Phoneme Synthesis and Library* window, where shapes can be tested with short synthesis examples and vocal tract

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 7. The initial SPASM screen, showing the windows that are opened upon running the program.



description files can be saved to disk. Figure 7e is the *Performance Feature Editor* window, which allows parameters affecting pitch to be controlled and saved to disk. Figure 7f is the *Diphone Synthesis and Library* window, which allows transitions between shape and glottal states to be specified, auditioned, and saved to disk. Hidden windows can be called up for more advanced control and analysis functions.

Vocal Tract Shape Control

The *Vocal Tract Editor* provides control over the shape of the acoustic tube (and thus the digital filter) which models the vocal tract. Shapes can be saved to or loaded from a shape library. Sliders in the graphical editor window control the radius of each segment of the tract. The path through the nasal airway is

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 8. Windows for controlling vocal tract and nasal tract shape.

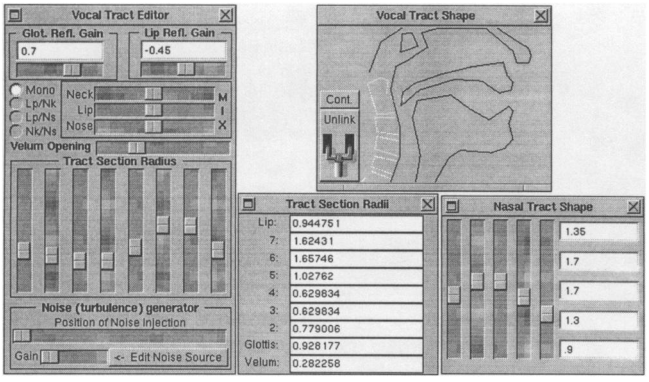
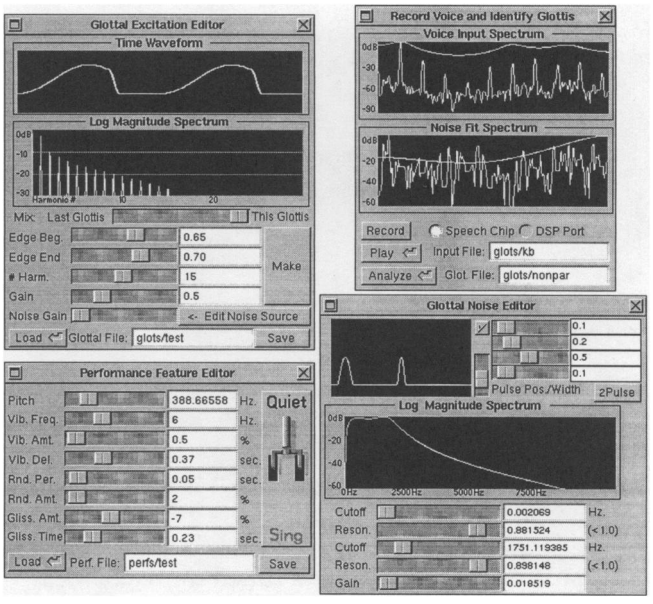


Fig. 9. Windows for controlling and identifying the glottal pulse source.



controlled by a velum position slider. A graphical cross section of a human head provides immediate feedback to the user about the vocal tract shape as controlled by the sliders or as loaded from a file. Another text window showing the radii in centimeters allows the user to enter parameters with greater accuracy. Yet another window allows the editing of the nasal tract shape parameters, although these characteristics are usually not varied in connected speech and singing. Switches and sliders control and mix the lip, nose, and throat radiation outputs. Figure 8 shows the vocal tract shape control windows.

The Glottal Source

The glottal source used in this system models the time-domain and spectral properties of the pressure waveform of the human glottis. The glottal source waveform is additively synthesized from Fourier coefficients controlled by simple parameters entered in the editor or from a library file of coefficients derived from analysis data. The simple parameter editor controls operate mainly on the time-domain glottal waveform. Parameters include the number of harmonics to be used for synthesis (primarily to prevent aliasing), the overall amplitude, and the position and slope of the falling edge of the glottal pulse, which has been shown to be an important feature when describing vocal effort (Rosenberg 1971; Sundberg 1987; Cummings and Clements 1990). Graphical displays of the log-magnitude spectrum and the time-domain waveform are provided. Parameters may be saved to and loaded from disk files. The *Glottal Noise Editor* allows the user to specify a time-domain pulse shape for an additive noise source, simulating the pulsating turbulence generated as the glottal folds open and close (Chafe 1990; Schumacher and Chafe 1990; Cook, Chafe, and Smith 1990; Cook 1991b, 1992). A default set of natural amplitude and frequency control functions are available for synthesis, or the user can edit the performance features using sliders or text fields. Performance controls affect glottal frequency parameters and are located in the *Performance Parameters* window. Figure 9 shows the glottal source editing and analysis windows and the performance controller window.

The Noise Source

Fricative consonants behave spectrally much like filtered noise, exhibiting one or two resonant peaks (Heinz and Stevens 1961; Shadle 1985), although the spectra often exhibit time-varying characteristics (Badin 1989; Cook 1991a). To simulate the turbulences of fricatives and other consonants, a noise

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 10. Windows for controlling and identifying the turbulent noise source.

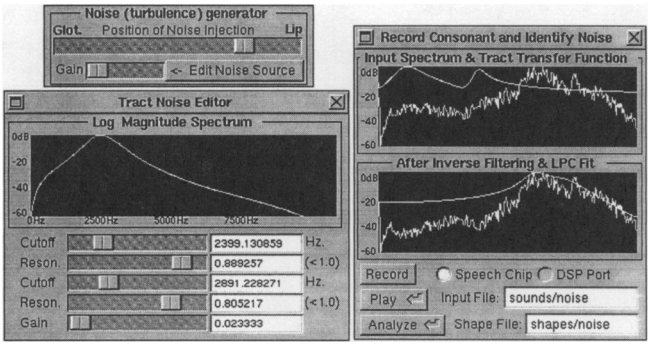


Fig.11. Two fricative consonant vocal tract shapes and power spectra of synthesized fricatives.

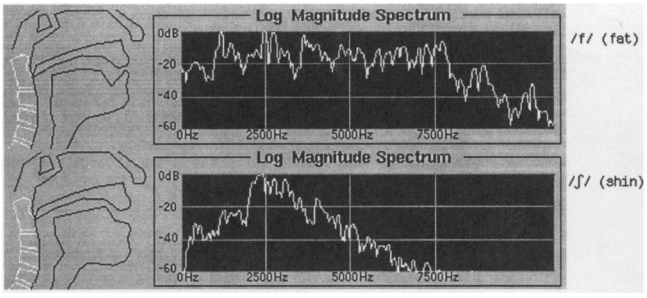
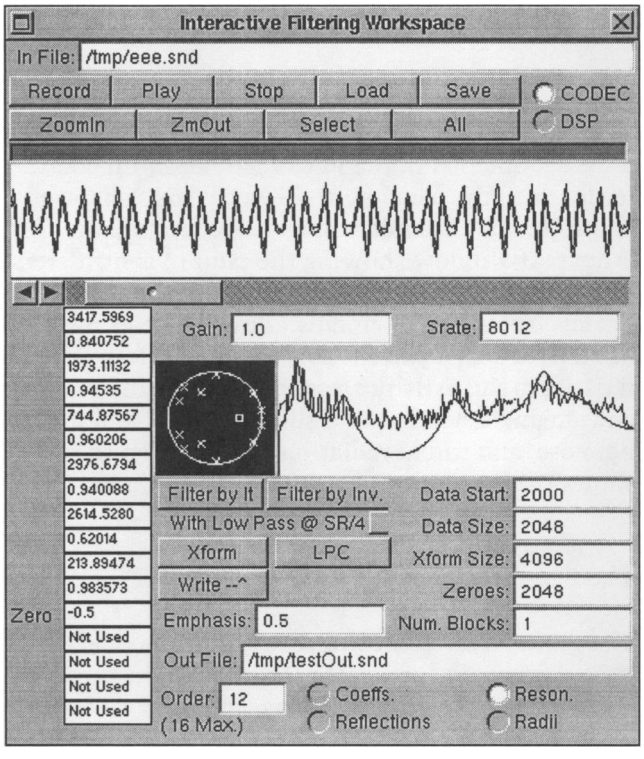


Fig. 12. An interactive filter editor, with controls for fitting, editing, and applying filters to sounds.



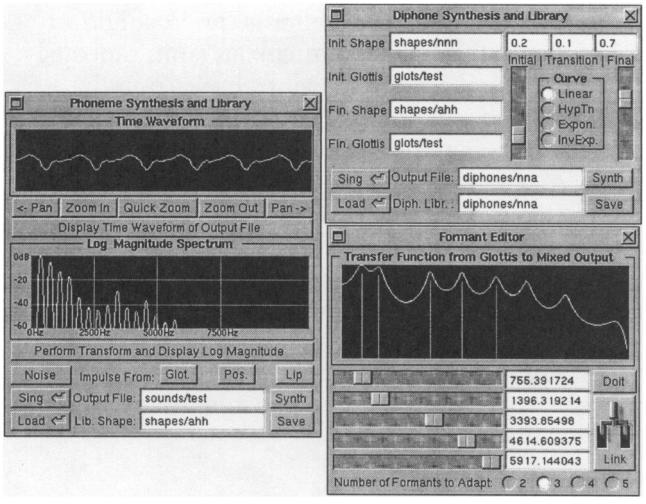
source can be placed at any point in the throat path of the vocal tract. The output of the noise source can be made arbitrarily resonant at two frequencies by a four-pole filter. This allows a tuned source of local turbulence to be injected at a point of constriction. Noise source parameters are saved as part of shape files. Figure 10 shows the noise controller windows. Figure 11 shows two vocal tract configurations for synthesizing fricatives, and the power spectra of the corresponding synthesized consonants.

Glottal Pulse and Noise Source Identification

To aid the user in obtaining natural source parameters quickly, and to provide a framework for detailed but rapid study and experimentation with glottal and noise sources, various identification/analysis functions are built into the system. The *Record Consonant and Identify Noise* window allows the user to record a fricative consonant or specify a prerecorded sound file. Using LPC, the system first inverse-filters any effects of the current vocal tract shape, then designs a four-pole filter that matches the spectrum of the inverse-filtered recorded sound. Similarly, the *Record Voice and Identify Glottis* window (shown in Fig. 9) allows the user to inverse-filter a recorded voice, using the current SPASM vocal tract configuration as the prototype. This inverse-filtering process yields an estimate of the glottal pulse function. An interactive inverse-filtering window is available for more accurate identification of the user's vocal tract transfer function and glottal input function. The user can fit an LPC filter to a sound, and the poles of the filter are located and displayed on a *Z*-plane view. The center frequencies and radius locations of the poles are displayed and can be edited by the user. Using this tool, interactive inverse-filtering of the vocal tract transfer function can be accomplished. Figure 12 shows the *Interactive Filtering Workspace.*

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 13. Phoneme and Diphone synthesis control windows, and the Formant Editor display window.



Phoneme and Diphone Synthesis

Once the glottal source, the noise source, and the vocal tract shape have been established, a short musical "performance" can be synthesized by clicking the mouse on the *Synth* button, in the *Phoneme Synthesis and Library* window (illustrated in Fig. 13). Once the synthesis is complete, the result can be heard via the NeXT computer's internal digital-to-analog converters. The file can be played back repeatedly by clicking the mouse on the *Sing* button. By typing any file name into the *Output File* field and pressing the carriage return, the file is played back (if it exists). Synthesis to different files allows speedy A/B comparison of sound examples. The time-domain waveform and the log-magnitude spectrum of the synthesized phoneme can be displayed.

Diphthongs can be constructed by specifying initial and final sets of parameters, an interpolation curve, and the time in seconds of the initial and final steady-state segments. The *Diphone Synthesis and Library* window (Fig. 13) controls these synthesis parameters. Since the synthesis yields 1 second of sound, specifying the duration of the initial and final states also specifies the transition time. Curves available for interpolation include linear, hyperbolic tangent, and exponentials. In the case of the glottis, linear interpolation is carried out between initial and final wavetables. For the noise generator, the filter pole parameters are interpolated in the Z plane. The vocal tract and nasal tract scattering relationships are interpolated in the radius space.

Formant Editor and Display

The *Formant Editor* window in Fig. 13 allows the user to display and edit features of the vocal tract filter in the formant domain (the one in which the ear perceives speech sounds). When this window is activated, the system impulse response is obtained. A log-magnitude transform is computed and displayed, and peaks are located and marked. Each of the first few (selectable) peaks is associated with a text-field/slider control, and the user may move the markers to new locations. By pressing the *Doit* button, the system adaptively moves the formant peaks to the desired locations, modifying the vocal tract in a "least squares perturbation" fashion. When the *Link* button is active, the formant display updates each time any change is made in vocal tract control parameters. This allows the complex connection between vocal tract controls and vocal tract filter spectra to be interactively explored.

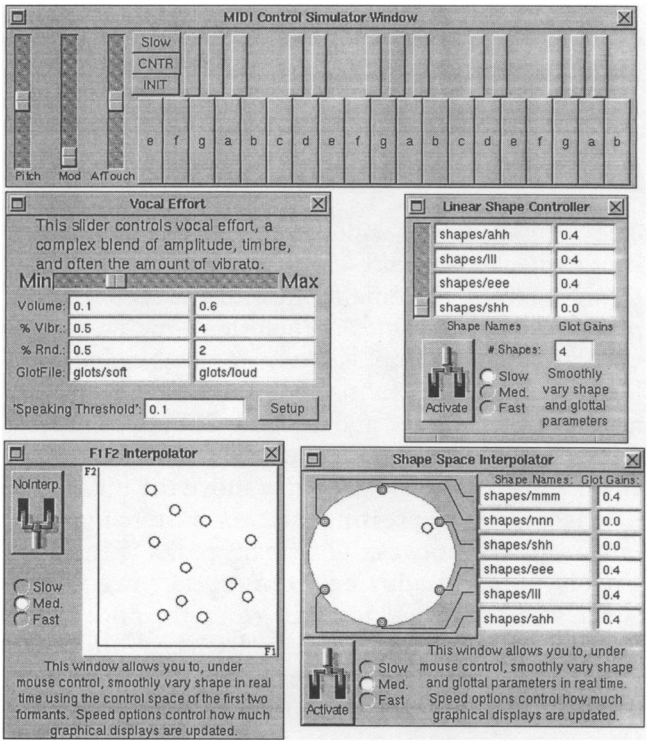
Real-Time DSP Synthesis and Control Strategies

By clicking the *Sing* switch in the *Performance Features* window (Fig. 9), the system begins to synthesize in real time. Performance features then active are pitch, vibrato speed and amount, and random vibrato amount. Tract section, velum, and noise controls modify the model and sound in real time. New glottis wavetables may be synthesized and downloaded to the DSP chip. A crossfader controls the mix between the current glottis wavetable and the last one.

The SPASM system is inherently a real-time DSP synthesis controller, but it quickly becomes clear that the computer keyboard and mouse are a rather impoverished interface for controlling such a complex synthesis model. A number of real-time controls have been developed for the SPASM model, allowing simple mouse and MIDI controls to make many modifications simultaneously to the vocal model pa-

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 14. Windows for controlling real-time performance and interpolation.



rameters. Figure 14 shows the control windows of several of the real-time performance controls. One interactive real-time tract shape controller is the *Shape Space Interpolator*. This window allows the user to enter a number of shape filenames into text fields. Each point along the edge of the circular shape space control area is a region that is dominated by a particular shape. The user may control the current vocal tract parameters by moving a cursor about the control area, thus determining the "mix" of shapes. The *F1F2 Vowel Space Interpolator* is similar in that the mouse position determines the mix of shapes, but the two-dimensional control surface is the plot of formant 1 versus formant 2, as is commonly found in the speech synthesis and perception literature (Peterson and Barney 1952). Another shape space interpolator is the *Linear Interpolator*, where all shapes are positioned along a line, and a single degree of freedom (slider or MIDI controller) controls shape.

Another real-time controller is the *Vocal Effort* controller, which in one dimension simultaneously controls glottal amplitude, glottal spectrum (by a crossfade between two previously loaded glottal files), and the amount of periodic and random vibrato. In this way, a single MIDI control such as aftertouch can be used to simulate the effort of the singer. A MIDI control window is provided to map MIDI controllers onto SPASM parameters. In the default mapping, note number maps to glottal pitch, portamento causes the singer to slide between pitches or move rapidly, aftertouch controls vocal effort, and modulation wheel controls the linear shape interpolator.

Singer: A Software Synthesis System

For many applications, accurate control of synthesis at the single-sample level is desired. Repeatable synthesis is important for generation of sounds for psychoacoustic testing and other investigations, and for composition. Toward these ends, a software synthesis system called "singer" based on the WGF model of the vocal tract was constructed. The program is written entirely in ANSI C, which simplifies porting it to other computers. The program takes as input a text file (actually a series of C function calls) that specifies targets for the model parameters and times for the transitions to take place. For speed of synthesis, linear interpolation is performed on all parameters. We will present the basic singer functions and arguments and an example singer control file.

The `synthesize` function is the heart of the singer program. This function interpolates the singer model from the last parameter values to a new set of parameter arguments, synthesizing and writing the samples to a sound file. The format of the `synthesize` function call is,

```
synthesize(time, FShape, FGlot,
  FPitch, FGlotAmp, FNoiseAmp,
  FVibrAmt, fd)
```

which interpolates the singer model from the last values to those given as arguments to the function call, synthesizing and writing the samples out to the file specified by the argument `fd`. The arguments are

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 15. A singer control file to sing the word "Shiela."

```
/************ Example Singer "Program" ******/
// This example sings the word "Shiela"
singer(int fd) {
// set thefile path to directories "shapes" and "glots"
   set_file_path("/local/prc/Library/SPASM/");
// initialize the singer model
   init();
// initial setup for performance
   setup("shh","soft",400.0,0.0,0.0,0.0);
//        time, shape,  glot,  freq., gAmp, nAmp, vibr.
   synthesize(0.3, "shh", "soft", 400.0, 0.0, 0.3, 0.00, fd);
   synthesize(0.1, "eee", "soft", 430.0, 0.2, 0.3, 0.04, fd);
   synthesize(0.7, "eee", "soft",  a4  , 0.2, 0.0, 0.07, fd);
   synthesize(0.2, "lll", "soft", 440.0, 0.4, 0.0, 0.04, fd);
   synthesize(0.2, "ahh", "soft", 400.0, 0.3, 0.0, 0.00, fd);
   synthesize(0.2, "ahh", "soft", 400.0, 0.3, 0.0, 0.00, fd);
   synthesize(1.5, "ahh", "loud", 400.0, 1.0, 0.0, 0.08, fd);
   synthesize(0.1, "ahh", "soft", 400.0, 0.0, 0.0, 0.08, fd);
   silence(0.5,fd);      // Write some silence
   return;
}
```

`time`, the time over which the current transition will take place;
`FShape`, a SPASM vocal tract shape file;
`FGlot`, a SPASM glottal parameter file;
`FPitch`, the pitch in hertz of the glottal source;
`FGlotAmp`, the amplitude of the glottal source;
`FNoiseAmp`, the amplitude of the fricative noise;
`FVibrAmt`, the amount of periodic vibrato of the glottal source; and
`fd`, a file descriptor of an open sound file.

Other parameters may be changed instantly by using the function calls:

`setPerfVibrFreq(float afreq)`, set the vibrato frequency in Hz;
`setPerfRndAmt(float r)`, set the random vibrato amount;
`setPerfRndPeriod(double r)`, set the random vibrato period;
`setGlotNoiseFilter(double gn_angle1, gn_rad1, gn_angle2, gn_rad2)`, set the noise filter parameters; and
`setPulseShape(double p1pos, p1width, p2pos, p2width, p2height, pfloor)`, set the noise pulse shape.

There are also several function calls to get current state of the SPASM model, as well as initialization functions, and other utility functions. Figure 15 demonstrates the use of singer, showing an annotated control "program" that synthesizes the word "Shiela." Note that in ANSI C program comments are either enclosed between "/*" and "*/" or are preceded by "//" and terminated by a carriage return.

Cantus Forma: Rules for Singing and Languages

LECTOR: Ecclesiastical Latin to Speech/Singing

For the SPASM and singer systems to be accessible to composers, singers, and other nontechnical users, it was desirable to create methods of data input that closely resemble the familiar representations of spoken words and singing. Such representations include simple text as well as note names and frequency values, and common music notation with text placed below the musical staff as it is in vocal music. A Latin text control language called LECTOR was created for controlling the SPASM/singer models (Cook 1991c).

The rules for the proper pronunciation of Ecclesiastical Latin are simple and rigid (Traupman 1966). Very few "exceptions" exist, and the context required to determine the pronunciation of a single character depends on at most two characters before or after the current character. This allows the parser to run linearly, processing the text input as a stream. The preservation of the stream control paradigm is considered important in this project, anticipating future systems that might pass the low-bandwidth text/control stream over a network or similar low-data-rate channel, for rendering of spoken or sung text at the other end of the transmission channel. The LECTOR input code and the corresponding singer control code generated by the LECTOR Latin parser for the word "Requiem" are shown in Fig. 16. Note the "rolled r" subroutine and the variables `vRef` and `cRef`, which allow time scaling of vowel and consonant times independently.

Any ANSI C command that the singer instrument understands may be imbedded into the text stream in curly braces. An example that might be used at the beginning of a performance is

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Fig. 16. A LECTOR text file and the resultant singer file output.

```
/****** LECTOR Text Control File ***********/
   Requiem
/********** End LECTOR Text ****************/

/****** Resultant Singer Control Code ************/
#include "singlib/singer.h"
#include "singroutines.h"
#include "tuning/tuning.h"
#define LG lastGlot()
#define LP lastPitch()
#define LV lastVibrAmt()

singer(int fd) {
double vRef=0.5,cRef=0.5;
setGlot("latindefault");
init();
set_file_path("/local/prc/Library/SPASM/");
setGlot("latindefault");
setup("rolledrc",LG,360.0,0.0,0.0,0.02);
synthesize(0.10*cRef,"rolledrc",LG,LP,0.00,0.00,LV,fd);
rollr(0.2*cRef,LG,LP,0.5,LV,fd);
synthesize(0.10*vRef,"ehh",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"ehh",LG,LP,0.80,0.00,LV,fd);
synthesize(0.05*cRef,"kkk",LG,LP,0.00,0.00,LV,fd);
synthesize(0.05*cRef,"kkk",LG,LP,0.00,0.00,LV,fd);
synthesize(0.02*cRef,"kk+",LG,LP,0.00,1.00,LV,fd);
synthesize(0.08*cRef,"kk+",LG,LP,0.00,0.20,LV,fd);
synthesize(0.10*vRef,"ooo",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"ooo",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"eee",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"eee",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"ehh",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"ehh",LG,LP,0.80,0.00,LV,fd);
synthesize(0.05*cRef,"mmm",LG,LP,0.80,0.00,LV,fd);
synthesize(0.15*cRef,"mmm",LG,LP,0.80,0.00,LV,fd);
synthesize(0.10*vRef,"mmm",LG,LP,0.00,0.00,LV,fd);
silence(0.1*vRef,fd);
return;
}
/********* End of Singer Control Code ************/
```

```
setClicking(0.4);
```

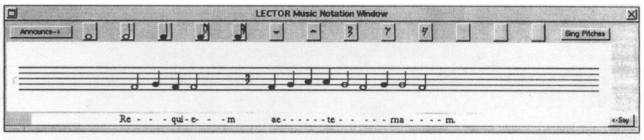
which causes the singer instrument to write a click into the output sound file each 0.4 seconds for metronome purposes. Pitches may be imbedded into the text stream by using <> symbols, such as

```
<f4>re <g2*pow(2.0,24.0/12.0)>qui
  <440.0>em.
```

All vowels have a default duration of `0.2*vRef`, but may be extended by `0.2*vRef` by placing a hyphen after the vowel, such as in the example:

```
<f4>re---<g4>qui-----<440.0>e----m.
```

Fig. 17. The LECTOR graphical control window and the resulting LECTOR text control file output.




```
/****** Resulting LECTOR Text Control File ******/
<f4>re--------<g4>---q<f4>ui---<f4>e-------m
<f4>ae---<g4>-----<a4>---t<a4>e---<g4>---------<f4>---------<g4>---
r<g4>na-------<f4>--------m.
/****** End LECTOR Text ********************/
```

A Graphical Controller for the LECTOR System

The LECTOR *Music Notation Window* provides a graphical data input mode for the Latin parser. The buttons at the top generate note symbols on the staff, which are then dragged with the mouse to the desired location. Text alignment with the note symbols determines duration, and the dashes in the text allow the parser to identify word continuations and boundaries. Figure 17 shows the graphical control window and the resulting LECTOR text control file output.

More Languages and Rules

Recently, a team consisting of the author and Dimitris Kamarotos, Akis Diamantopoulos, and Giorgos Filippis of the Center for Contemporary Music Research in Athens, Greece, implemented a system for the Greek language based on the LECTOR system. Many new features have been added to the system, including tuning systems for several microtonal Greek modes and subroutines for common Greek vocal ornaments.

Conclusions

A number of examples of singing synthesis systems have been built using the SPASM, singer, LECTOR, and Greek language programs. The combination of real-time DSP control and software synthesis allows the user to experiment quickly, yet produce repeatable results. Many syntheses yield extremely natural sounds on the first attempt. Since descriptions of un-

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

natural-sounding synthesized sounds often rely on physical references ("She sounds like her jaw is open too far," or "His tongue sounds fat"), the physical parameters indicate what to do to the model controls if the synthesis does not sound correct. The programs have been made available to composers for use in musical compositions and to psychologists for use in the generation of stimuli for psychoacoustic testing. They can be retrieved using anonymous Internet ftp file transfer from the files named SPASM.tar, and singer.tar on the server ccrma-ftp.stanford.edu. The real-time control possibilities show great promise, and given suitable controllers and more advanced control-to-parameter mappings, SPASM could become a new instrument for expressive real-time performance.

Acknowledgments

Research on this project was carried out under grants from the CCRMA Industrial Affiliates program and DYNACORD, Inc. The Greek singer project was funded by the Greek Ministry of Culture and the Center for Contemporary Music Research (CCMR), Athens. Thanks to Thanassis Rikakis and everyone else at CCMR for making the Greek project possible. I would like to thank as always Julius O. Smith, John Chowning, Chris Chafe, and everyone else at CCRMA for their support in this work.

References

Atal, B. S. 1970. "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave." *Journal of the Acoustical Society of America* 47:65(A).

Badin, P. 1989. "Acoustics of Voiceless Fricatives: Production Theory and Data." *Speech Transmission Laboratory, Quarterly Progress and Status Report.* STL-QPSR 3. Stc :kholm, Sweden: Royal Institute of Technology, pp. 33–55.

Chafe, C. D. 1990. "Pulsed Noise in Self-Sustained Oscillations of Musical Instruments." In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP,* vol 2(S2). New York: Institute of Electrical and Electronics Engineers, pp. 1157–1160.

Chowning, J. M. 1989. "Frequency Modulation Synthesis of the Singing Voice." In M. V. Mathews and J. R. Pierce, eds. *Current Directions in Computer Music Research.* Cambridge, Massachusetts: MIT Press, pp. 57–63.

Cook, P. R., C. D. Chafe, and J. O. Smith. 1990. "Pulsed Noise in Musical Systems, Techniques for Extraction, Analysis, and Visualization." In *Proceedings of the 1990 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 63–65.

Cook, P. R. 1991a. "Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing." Stanford Publication STAN-M-68. Electrical Engineering Ph.D dissertation, Stanford University, Stanford Center for Computer Research in Music and Acoustics.

Cook, P. R. 1991b. "Noise and Aperiodicity in the Glottal Source: A Study of Singer Voices." In *Twelfth International Congress of Phonetic Sciences.* Aix-en-Provence, France: University of Provence Publication Services.

Cook, P. R., 1991c. "LECTOR: An Ecclesiastical Latin Control Language for the SPASM/singer Instrument." In *Proceedings of the 1991 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 319–321.

Cook, P. R. 1992. "Aperiodicities in the Singer Voice Source." In *Proceedings of the Fall 1991 Acoustical Society of America Conference.* New York: Acoustical Society of America.

Cummings, K. E., and M. A. Clements. 1990. "Analysis of Glottal Waveforms Across Stress Styles." In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP,* vol 1(S1). New York: Institute of Electrical and Electronics Engineers, pp. 369–372.

Heinz, J. M., and K. N. Stevens. 1961. "On the Properties of Voiceless Fricative Consonants." *Journal of the Acoustical Society of America* 33: 589–596.

Kelly, J. L., and C. C. Lochbaum. 1962. "Speech Synthesis." In *Proceedings of the Fourth International Congress on Acoustics* Paper G42: pp. 1–4.

Markel, J. D., and A. H. Gray. 1976. *Linear Prediction of Speech.* New York: Springer Verlag.

McAulay, R. J., and T. F. Quatieri. 1986. "Speech Analysis/Synthesis Based on a Sinusoidal Representation." *IEEE Transactions on Acoustics Speech and Signal Processing* ASSP-34(4):744–754.

Peterson, G. E., and H. L. Barney. 1952. "Control Methods Used in a Study of the Vowels." *Journal of the Acoustical Society of America* 24(2): 175–184.

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions

Rodet, X. 1984. "Time-Domain Formant Wave-Function Synthesis." *Computer Music Journal* 8(3): 9–14.
Rosenberg, A. E. 1971. "Effect of Glottal Pulse Shape on the Quality of Natural Vowels." *Journal of the Acoustical Society of America* 49(2) Part 2: 583–590.
Schumacher, R. T., and C. D. Chafe. 1990. "Detection of Aperiodicity in Nearly Periodic Signals." In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP,* vol 2(S2). New York: Institute of Electrical and Electronics Engineers, pp. 1161–1164.
Serra, X., and J. O. Smith III. 1990. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition." *Computer Music Journal* 14(4): 12–24.
Shadle, C. H. 1985. "The Acoustics of Fricative Consonants." Ph.D. dissertation, Massachusetts Institute of Technology Research Laboratory of Electronics, Cambridge.
Smith, J. O. 1987. "Musical Applications of Digital Waveguides." Stanford Publication STAN-M-39. Stanford University Center for Computer Research in Music and Acoustics.
Smith, J. O. 1992. "Physical Modeling Using Digital Waveguides." *Computer Music Journal* 16(4).
Sundberg, J. 1987. *The Science of The Singing Voice.* Dekalb, Illinois: Northern Illinois University Press.
Traupman, J. C. 1966. *The New College Latin and English Dictionary.* New York: Bantam Books, NY.

This content downloaded from 129.174.21.5 on Mon, 9 Sep 2013 15:36:04 PM
All use subject to JSTOR Terms and Conditions